Report II

Miguel Rodríguez, RA: 192744, Email: m.rodriguezs1990@gmail.com

I. Problem

The objective of this work is to study the different basic techniques to be able to perform video summarization The video summarization is a technique used to make a visual understanding of the frames of video, which consists of obtaining the abrupt transitions within the video, and summarize the scenes that occur in a video. It can be seen in Fig. 1, as a video can be decomposed into scenes and each of those scenes in shots, which mark the abrupt transactions that exist between one frame and another. The most classic techniques of video summarization are able to capture these shots changes and to make the summary from each of these abrupt changes. This reduction of the video can help in different areas that are used the videos, such as: video surveillance, where it is necessary to make rapid reviews of the events in the videos; in video indexing, where it is important to be able to make a compression of the information of the video; in pages where the content shown is based on videos, which would allow you to see a visual summary of what is shown in a video before playing it; and finally can be used in machine learning techniques to not process the complete video, the processing only the summary. This work will be divided into four parts, which will be the application of four techniques to be able to perform video summarization.

A. Pixels difference

This technique consists in finding the shots of a video through the difference that exists between the pixels of the frames. In order to calculate these differences a simple subtraction is perfomed between the analyzed frame and the predecessor frame, if the number of pixels that are greater that a threshold t_1 is greater that another threshold t_2, then that frame is considered as a abrupt change.

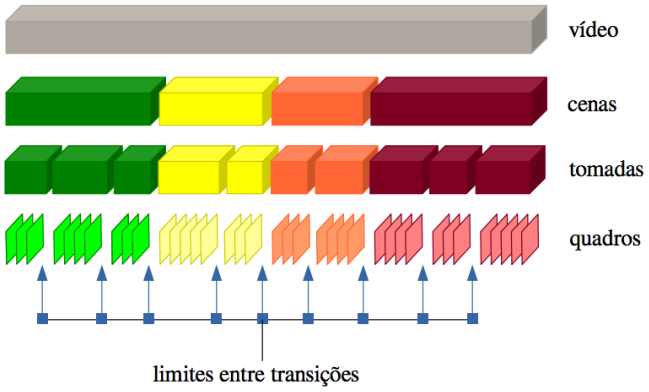


Fig. 1: Decomposition of a video, in scenes, shots and frames.

B. Blocks difference

This technique consists in finding the shots of a video through the decomposition of the frames in $N \times N$ blocks. Where the difference or error between the analyzed frame and its predecessor is calculated, if the error is greater that a threshold t_1 and the number of blocks that exceeds that error is greater than t_2, the frame is considered a new shot.

C. Histograms difference

This technique consists in finding the shots of a video through the difference of histograms, if the sum of the histograms difference is grater that the threshold t_1, the frame is considered a new shot.

D. Edges difference

This technique consist in finding the shots of a video through the existing differences in the border maps of the video frames. If the differences is greater that a threshold t_1, the frame is considered a new shot.

II. Solution

The video summarization is a technique used to summarize events that occur in a video, these events can be cataloged as shots (See Fig. 1). In order to find these shots in a video, it is necessary to analyze frame by frame to see when sudden changes occur, which are marked as a frame change. In this work, we present four different parametric techniques that enable the recognition of key frames for video summarization.

A. Pixels difference

This technique can be considered as the most naive approach, consists in performing a scan of the video frame by frame, where each frame t_n is analyzed with its predecessor t_{n-1}. In order to decide whether a frame is considered a new shot, the following formulation is used: the subtraction of the current frame is performed with the previous one, after this resulting image, all pixels that are greater that a threshold t_1 (Which varies among the possible values of the gray scale, for all experiment between 0-255) are counted, and this count is divided by the number of pixels of the frame (to be able to have numbers between 0 and 1), if this account is greater that a threshold t_2, the frame is considered the beginning of a new shot, and this is considered within the summary of the video. This process can be seen as a code in the Listing 1.

```
1 def shot_cut_pixel_differences(frame0, frame1,
      pixel_threshold, frame_threshold):
2  #Get difference
3  diff = cv2.absdiff(frame0,frame1)
4  diff[diff < pixel_threshold] = 0
```

```
diff[diff >= pixel_threshold] = 1
#Get size of frames
size = 1
for x in diff.shape:
 size *= x
ratio = diff.sum()/size
#Verify threshold
if (ratio >= frame_threshold):
 return True, ratio
else:
 return False, ratio
```

Listing 1: Function that calculates whether a frame and its precedessor have a difference that can be considered as a change of shot.

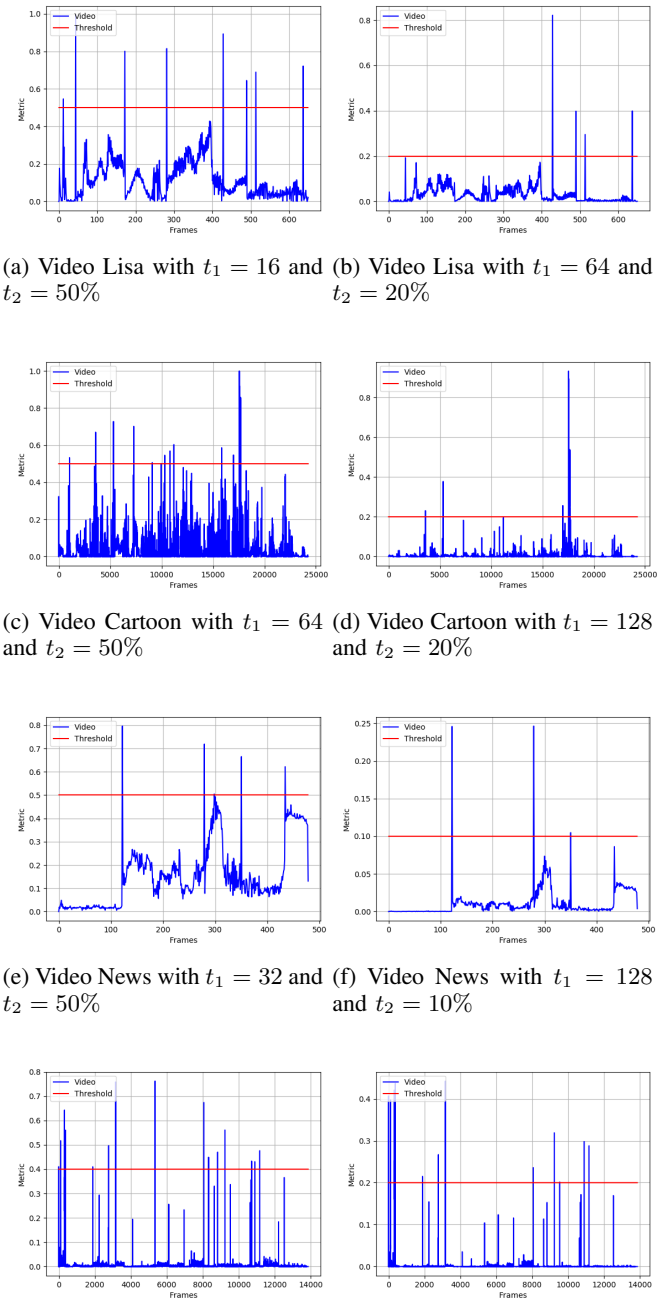
For this procedure we used four videos for experiments. These were to perform a trial and error search of the best hyper parameters for thresholds t_1 and t_2, in order to perform this search we use the graphs of difference shown in Fig. 2, in which we can see how the difference between the frames throughout the video (Blue line) behave and the cut threshold is shown in red to choose which frames are chosen for the summarization. As we can seen in the graphs, finding a parameter that allows to optimize the number of shots of a video is a process that entails experimentation for each video. In Figs. 2a, 2b, 2e and 2f, which belong to the video Lisa and the video News, we can see that it was very easy to find a threshold, this because the frame that mark the transition, have very abrupt difference, for it was easy to find the frames to summarize the video. This can be seen in the Fig. 3a, 3b, 3e and 3f, in which summaries resulting from the extraction of key frames are shown.

On the other hand, in Figs. 2c, 2d, 2g and 2h, we can see that finding the key frames for transition is very difficult, because these videos are longer and show many transitions between them, they also have frames with the same shape, but they have different gray scale values. For this method to have different gray scale values if the differences is very large can be counted as a key frame, being that it stays in the same scene. In Figs. 3c, 3d, 3g and 3h, we can see that the resulting summaries of the videos have many frames selected as scene changes, but these are part of this scene, this becaus the color changes cause the method to make the error to classify.

In conclusion, the method of difference of pixels is a naive way of attacking the problem of video summarization, this one has quite acceptable solutions, but the method has two big problems, first the dependence of the colors shown, and the parameterization of the variables. This method well parameterized could be used to make summaries of videos where it is known that the environment variables are well controlled.

B. Blocks difference

This technique is a little more elaborate that the technique previously shown, it also consists of calculating the difference between each frame of the video and through a procedure know if the frame analyzed correspond to a transition or not. In order to know if a frame is a change of shot, each frame is splitted into $N \times N$ smaller images, which are compared to each other



(a) Video Lisa with $t_1 = 16$ and $t_2 = 50\%$ (b) Video Lisa with $t_1 = 64$ and $t_2 = 20\%$

(c) Video Cartoon with $t_1 = 64$ and $t_2 = 50\%$ (d) Video Cartoon with $t_1 = 128$ and $t_2 = 20\%$

(e) Video News with $t_1 = 32$ and $t_2 = 50\%$ (f) Video News with $t_1 = 128$ and $t_2 = 10\%$

(g) Video RCA with $t_1 = 32$ and $t_2 = 40\%$ (h) Video RCA with $t_1 = 64$ and $t_2 = 20\%$

Fig. 2: Resulting graphs from the application of the pixel difference process. The x axis shows the frames and the y axis show the metric difference used to categorize whether or not a frame is a change of shot.

(a) Video Lisa with $t_1 = 16$ and $t_2 = 50\%$

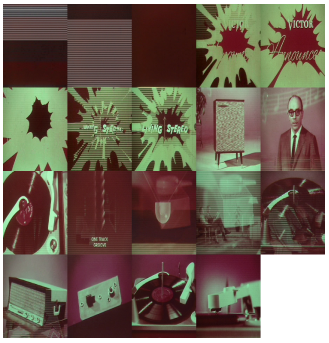
(b) Video Lisa with $t_1 = 64$ and $t_2 = 20\%$

(c) Video Cartoon with $t_1 = 64$ and $t_2 = 50\%$

(d) Video Cartoon with $t_1 = 128$ and $t_2 = 20\%$

(e) Video News with $t_1 = 32$ and $t_2 = 50\%$

(f) Video News with $t_1 = 128$ and $t_2 = 10\%$



(g) Video RCA with $t_1 = 32$ and $t_2 = 40\%$

(h) Video RCA with $t_1 = 64$ and $t_2 = 20\%$

Fig. 3: Summaries of selected frames, in that we can see the summaries produced by the application of the Listing 1.

	Experiments					
	I			II		
Video	t_1	t_2	Frames	t_1	t_2	Frames
Lisa	32	50%	3	64	30%	2
Cartoon	64	50%	29	128	20%	35
News	32	50%	5	128	10%	3
RCA	32	40%	18	64	20%	24

TABLE I: Summary table of the experiments performed with the videos, show the thresholds and number of frames found that satisfy the restrictions.

through the use of the Normalized Mean Square Error (See eq. 1). A frame is marked as a new shot when the sum of all splitted images where the calculated error is greater that t_1 is greater than another threshold t_2. Where t_1 is the % of error accepted to differentiate each tile, and t_2 is the percentage of small tiles in the image that are marked as different needed to mark the frame as different.

$$NMSE(F_n, F_{n+1}) = \sum \frac{(F_n - F_{n+1})^2}{F_n^2} \quad (1)$$

In order to divide the image into $N \times N$ tiles, we used the function created in the work zero of the classroom, which, given a size N o n_tiles, returns a vector with $N \times N$ images belonging to the original image. This function can be seen in the Listing 2.

```
def tiled(img, n_tiles):
 tiles = []
 height, width = img.shape

 for i in range(0,n_tiles):
  for j in range(0,n_tiles):
   row_range_lower = (i * height)//n_tiles
   row_range_upper = (((i + 1) * height)//n_tiles) -
     1
   col_range_lower = (j * width)//n_tiles
   col_range_upper = (((j + 1) * width)//n_tiles) -
    1
   tiles.append(img[row_range_lower:row_range_upper,
    col_range_lower:col_range_upper])
 return tiles
```

Listing 2: Function used to split image into $N \times N$ tiles.

After dividing the observed frame and its predecessor, it is necessary to know if it meets the requirements to be considered a key frame, so for this process we use the function of Listing 3, which calculates the error through Eq 1 for each block difference, and counts which are greater that t_1, then see if the number of blocks that gave true are grater that t_2, if so, that frame is considered as a key frame.

```
def shot_cut_blocks(blocks0, blocks1, n_tiles,
    error_threshold, blocks_threshold):
 diff_blocks = 0
 for i in range(n_tiles*n_tiles):
  nmse = error(blocks0[i], blocks1[i])
  if(nmse >= error_threshold):
   diff_blocks += 1
 diff_blocks /= (n_tiles**2)
 if(diff_blocks >= blocks_threshold):
  return True, diff_blocks
 else:
```

	Experiments							
	I				II			
Video	n_tiles	t_1	t_2	Frames	n_tiles	t_1	t_2	Frames
Car	8	90%	60%	4	16	90%	60%	3
Lisa	8	90%	70%	3	32	50%	70%	11
Cartoon	8	90%	80%	14	16	90%	90%	1
News	8	90%	70%	5	16	90%	70%	4
Umn	8	40%	18%	2	16	40%	18%	2

TABLE II: Summary table of the experiments performed with the videos, show the thresholds and number of frames found that satisfy the restrictions.

```
11    return False, diff_blocks
```

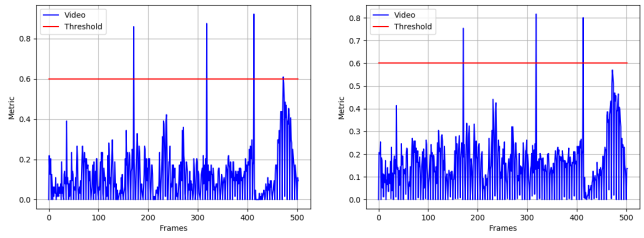
Listing 3: Function used to detect whether a frame is a shot change.

The big problem of the methods with hyper parameters, is the problem of the optimization of the parameters for its better operation. As in all the solutions proposed in this work, for this method it was also necessary to carry out an exhaustive search of the parameters for a better performance. In Fig. 4, the graphs resulting from search process are shown, in which the videos such as Car, Lisa and News (See Figs. 4a, 4b, 4c, 4d, 4g and 4h) don't present any problem when easily finding the hyper parameters. On the other hand, in the Cartoon video (The same video that presented problems with the previous method) presents problems to be able to find the key frames, this due to the great amount of changes that has. The method of blocks being a more robust method, has a better behavior with respect to the changes of colors in the same scene, as can be seen in the summaries deployed in Fig. 5, The video Cartoon no longer presents the problem of the choice of key frame of the same scene only by the changes of tones. But this problem of tones still persists in this method, as can be seen in the summary of the Lisa video (See Fig. 5c) where it is observed that in the same scene, several frames were chosen as key frames, this due to the change of contrast in the scene. The results of the hyper parameter search for these videos can be seen in the Table II, which shows the values chosen for the hyper parameters and the number of key frames found.

As a conclusion we can say that this method is much more robust that the previous one, because the differences do not measure them through the whole image, but it divides the problem into measuring the difference between several small sub images, while more bigger the division, more detail can be had, but in turn the method becomes slower due to the amount of calculations that should be performed. Also the use of a normalized error metric allows to make comparisons between images in percentage form, which allows to see better what is being done.

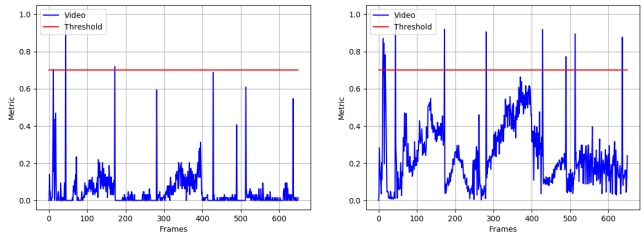
C. Histograms difference

This technique make use of the histogram tool, which can be explained as the visual representation of the probability distribution of each bin of an image, where each bin can be one or more gray scale values. The technique consist of making the



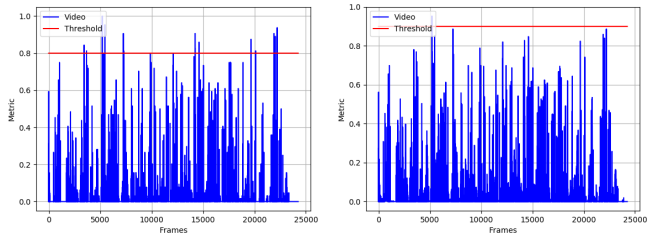
(a) Video Car with $n\_tiles = 8$, $t_1 = 90\%$ and $t_2 = 60\%$

(b) Video Car with $n\_tiles = 16$, $t_1 = 90\%$ and $t_2 = 60\%$



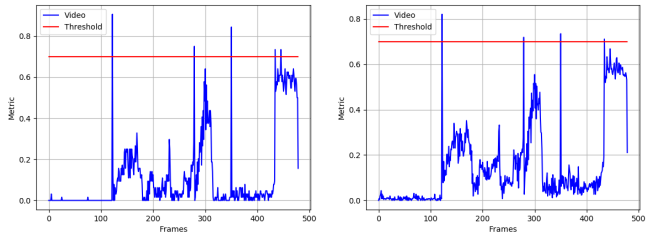
(c) Video Lisa with $n\_tiles = 8$, $t_1 = 90\%$ and $t_2 = 70\%$

(d) Video Lisa with $n\_tiles = 32$, $t_1 = 50\%$ and $t_2 = 70\%$



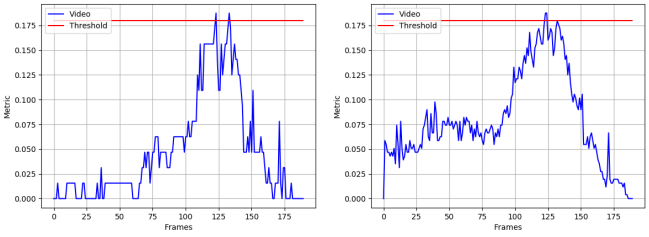
(e) Video Cartoon with $n\_tiles = 8$, $t_1 = 90\%$ and $t_2 = 80\%$

(f) Video Cartoon with $n\_tiles = 16$, $t_1 = 90\%$ and $t_2 = 90\%$



(g) Video News with $n\_tiles = 8$, $t_1 = 90\%$ and $t_2 = 70\%$

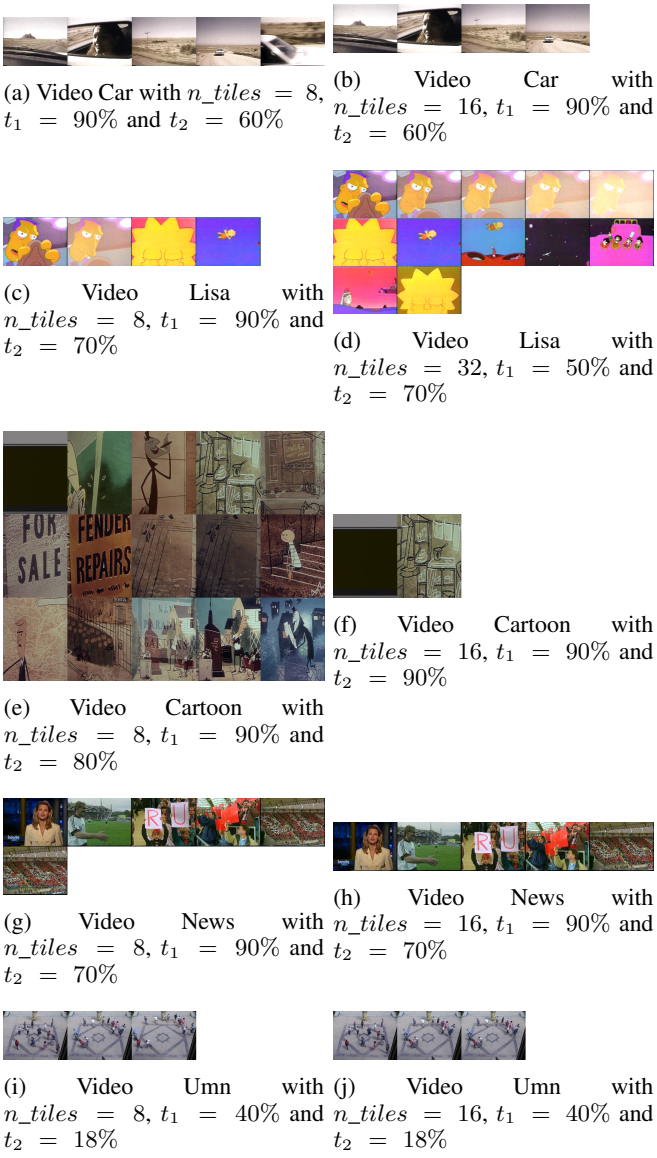
(h) Video News with $n\_tiles = 16$, $t_1 = 90\%$ and $t_2 = 70\%$



(i) Video Umn with $n\_tiles = 8$, $t_1 = 40\%$ and $t_2 = 18\%$

(j) Video Umn with $n\_tiles = 16$, $t_1 = 40\%$ and $t_2 = 18\%$

Fig. 4: Resulting graphs from the application of the block difference process. The x axis shows the frames and the y axis show the metric difference used to categorize whether or not a frame is a change of shot.



(a) Video Car with $n\_tiles = 8$, $t_1 = 90\%$ and $t_2 = 60\%$

(b) Video Car with $n\_tiles = 16$, $t_1 = 90\%$ and $t_2 = 60\%$

(c) Video Lisa with $n\_tiles = 8$, $t_1 = 90\%$ and $t_2 = 70\%$

(d) Video Lisa with $n\_tiles = 32$, $t_1 = 50\%$ and $t_2 = 70\%$

(e) Video Cartoon with $n\_tiles = 8$, $t_1 = 90\%$ and $t_2 = 80\%$

(f) Video Cartoon with $n\_tiles = 16$, $t_1 = 90\%$ and $t_2 = 90\%$

(g) Video News with $n\_tiles = 8$, $t_1 = 90\%$ and $t_2 = 70\%$

(h) Video News with $n\_tiles = 16$, $t_1 = 90\%$ and $t_2 = 70\%$

(i) Video Umn with $n\_tiles = 8$, $t_1 = 40\%$ and $t_2 = 18\%$

(j) Video Umn with $n\_tiles = 16$, $t_1 = 40\%$ and $t_2 = 18\%$

Fig. 5: Summaries of selected frames, in that we can see the summaries produced by the application of the Listing 3.

difference between the histograms of the current frame and the previous frame, if the sum of the differences (represented in the eq. 2) is greater than the threshold T, the frame is considered as a transition. To calculate the threshold t it is necessary to use the eq. 3, where μ is the mean of all differences during the video and σ is the standard deviation of them, so this method is only parameterized to the variable α and to the size of histogram.

$$D_i = \frac{\sum_{j=1}^{nBins} |H_{i-1}(j) - H_i(j)|}{2} \quad (2)$$

$$T = \mu + \alpha\sigma \quad (3)$$

The histograms used for this metric are normalized (obtained by means of the function in the Listing 4), for the eq. 2, is also normalized, so all results obtained vary between 0 and 1.

```
def normalized_histogram(image, nbins):
 if len(image.shape) == 3:
  height, width, channels = image.shape
 else:
  height, width = image.shape
  channels = 1

 histr = []
 for i in range(channels):
  hist = cv2.calcHist([image],[i],None,[nbins
    ],[0,256]).flatten()
  hist = hist/(height * width)
  histr.append(hist)

 return np.array(histr)
```

Listing 4: Function used to obtain the normalized histogram of an image

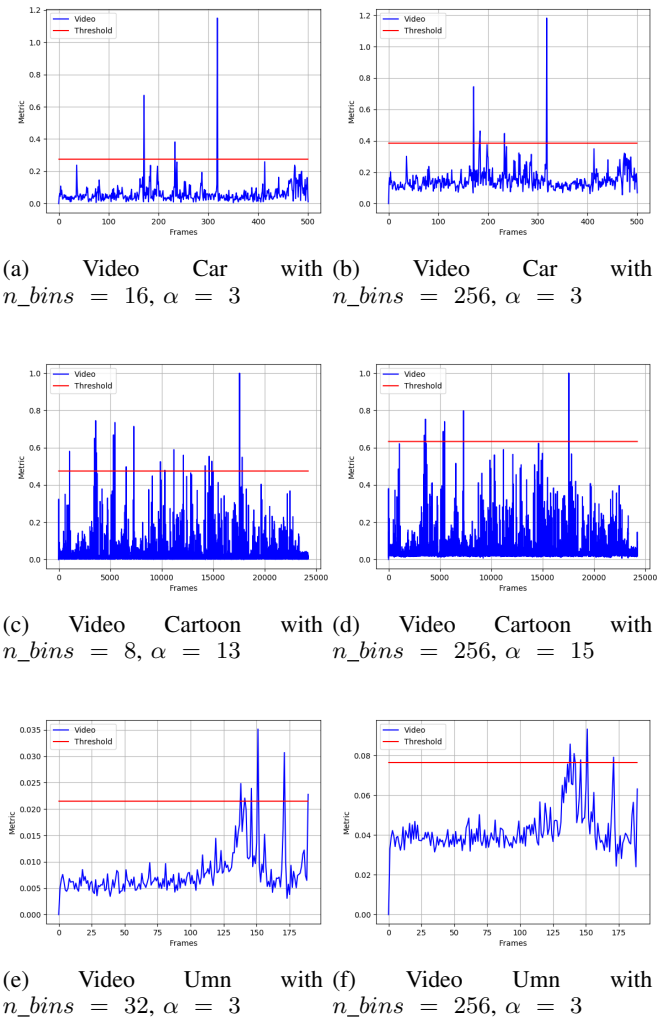
For this method experiments were performed with three videos, which as in previous methods were used to find the optimum hyper parameters that allowed better obtaining the key frames. The final results with the calculated hyper parameters can be observed in the Table III, also we can see the graphical differences shown in Fig. 6, In which it is shown that the video Car (See Fig. 6a and 6b) shown significant differences when it comes to being able to see which are the key frames, but seeing the results of extracting those frames, it can be seen that the extracted frames are not as representative of the video (See 7a and 7b). On the other hand, the video Umn (See Fig. 6e) and 6f), as in all other method, its necessary to make a search of the changes, this because all the actions occur in the same scene, for which the results of the summarized video are very poor (See. 7f and 7e)). Finally the Cartoon video, being a very long video and with many transitions also shows problems to be summarized, so this method does not show much help or difference with respect to the previous ones.

D. Edges difference

This technique is the most advanced of all techniques presented in this work, because it is based on the differences between the edges of the current frame and the previous frame, which makes it a bit more robust when leaving out the values

	Experiments					
	I			II		
Video	n_bins	α	Frames	n_bins	α	Frames
Car	16	3	3	256	3	5
Cartoon	8	13	18	256	15	9
Umn	32	3	6	256	3	5

TABLE III: Summary table of the experiments performed with the videos, show the thresholds and number of frames found that satisfy the restrictions.



(a) Video Car with $n\_bins = 16, \alpha = 3$

(b) Video Car with $n\_bins = 256, \alpha = 3$

(c) Video Cartoon with $n\_bins = 8, \alpha = 13$

(d) Video Cartoon with $n\_bins = 256, \alpha = 15$

(e) Video Umn with $n\_bins = 32, \alpha = 3$

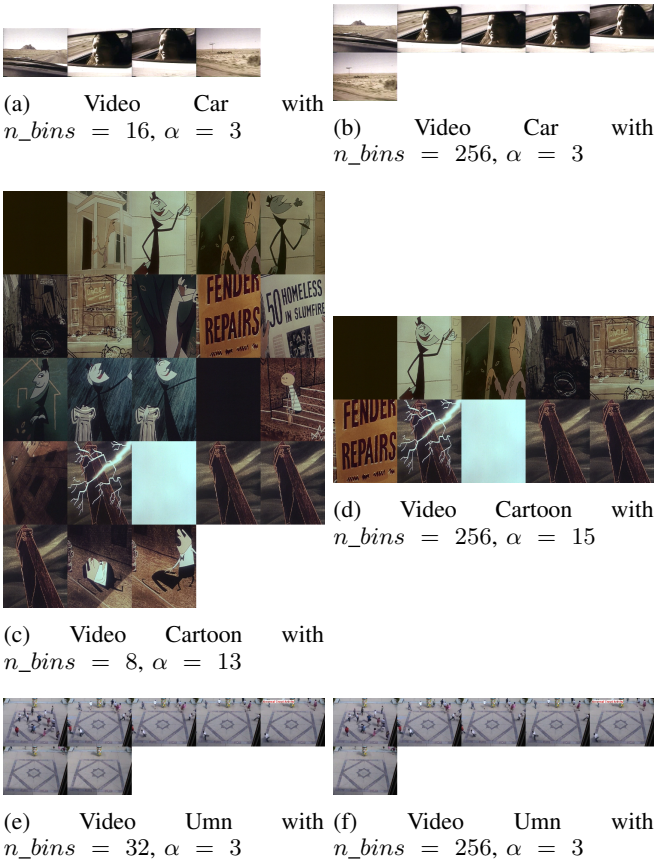
(f) Video Umn with $n\_bins = 256, \alpha = 3$

Fig. 6: Resulting graphs from the application of the histogram difference process. The x axis shows the frames and the y axis show the metric difference used to categorize whether or not a frame is a change of shot.

in gray scale and check the shape of objects to see if there is an abrupt transition. In order to define if a frame is a key frame, it is necessary to extract the normalized difference between the border map of both frames (See eq. 4), and contrast them against a threshold t_1, if the difference is greater, the current frame is considered a transition, this process can be seen in the Listing 5.

$$ratio = \frac{\sum |edges_{n-1} - edges_n|}{\sum edges_{n-1} + edges_n} \quad (4)$$

In order to obtain the edges of each frame, we used the Canny border detector, which has two hyperparameters, which for all experiment took the value of 255 and 85, this values are recommended by the creator of the method.



(a) Video Car with $n\_bins = 16, \alpha = 3$

(b) Video Car with $n\_bins = 256, \alpha = 3$

(c) Video Cartoon with $n\_bins = 8, \alpha = 13$

(d) Video Cartoon with $n\_bins = 256, \alpha = 15$

(e) Video Umn with $n\_bins = 32, \alpha = 3$

(f) Video Umn with $n\_bins = 256, \alpha = 3$

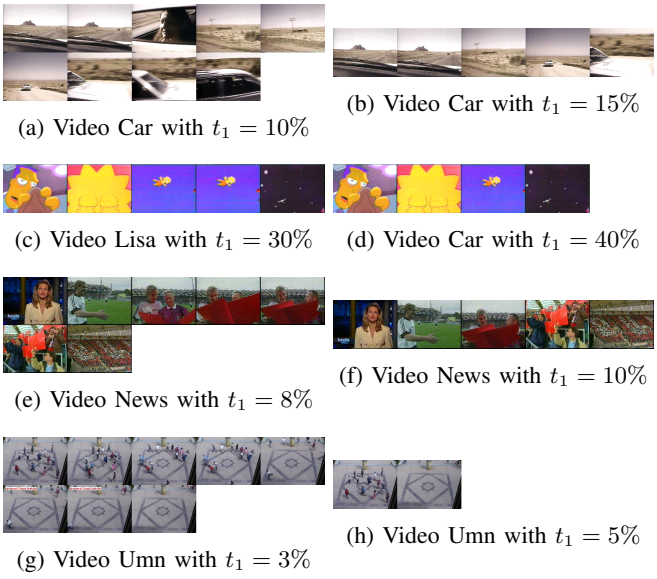
Fig. 7: Summaries of the selected frames as scene changes, in that you can see the summaries produced by te application of the method histogram difference.

```
def shot_cut_edges(edge0_sum, edge1_sum, threshold):
 ratio = 0
 if(edge0_sum+edge1_sum != 0):
  ratio = np.abs(edge0_sum - edge1_sum)/(edge0_sum+
    edge1_sum)

 if(ratio > threshold):
  return True, ratio
 else:
  return False, ratio
```

Listing 5: Function used to know if a frame is an abrupt transition

For this method, experiments were performed with five videos, which can be observed in the Table IV, for which the threshold t_1 is a value that is between 0% and 100%. The graphical results of these experiments can be seen in Fig. 9, in which it is seen that the behavior of this method with the videos easier to summarize is the desired one (See Figs. 9a, 9b, 9c 9d, 9g and 9h). On the contrary, and as with the other methods of work, the videos Cartoon and Umn have a very small performance. The video Cartoon being a very long video this method finds many false positives, which thanks to



(a) Video Car with $t_1 = 10\%$ (b) Video Car with $t_1 = 15\%$

(c) Video Lisa with $t_1 = 30\%$ (d) Video Car with $t_1 = 40\%$

(e) Video News with $t_1 = 8\%$ (f) Video News with $t_1 = 10\%$

(g) Video Umn with $t_1 = 3\%$ (h) Video Umn with $t_1 = 5\%$

Fig. 8: Summaries of the selected frames as scene changes, in that you can see the summaries produced by te application of the method histogram difference.

	Experiments			
	I		II	
Video	t_1	Frames	t_1	Frames
Car	10%	8	15%	4
Lisa	30%	4	40%	3
Cartoon	90%	97	99%	85
News	8%	6	10%	4
Umn	3%	7	5%	1

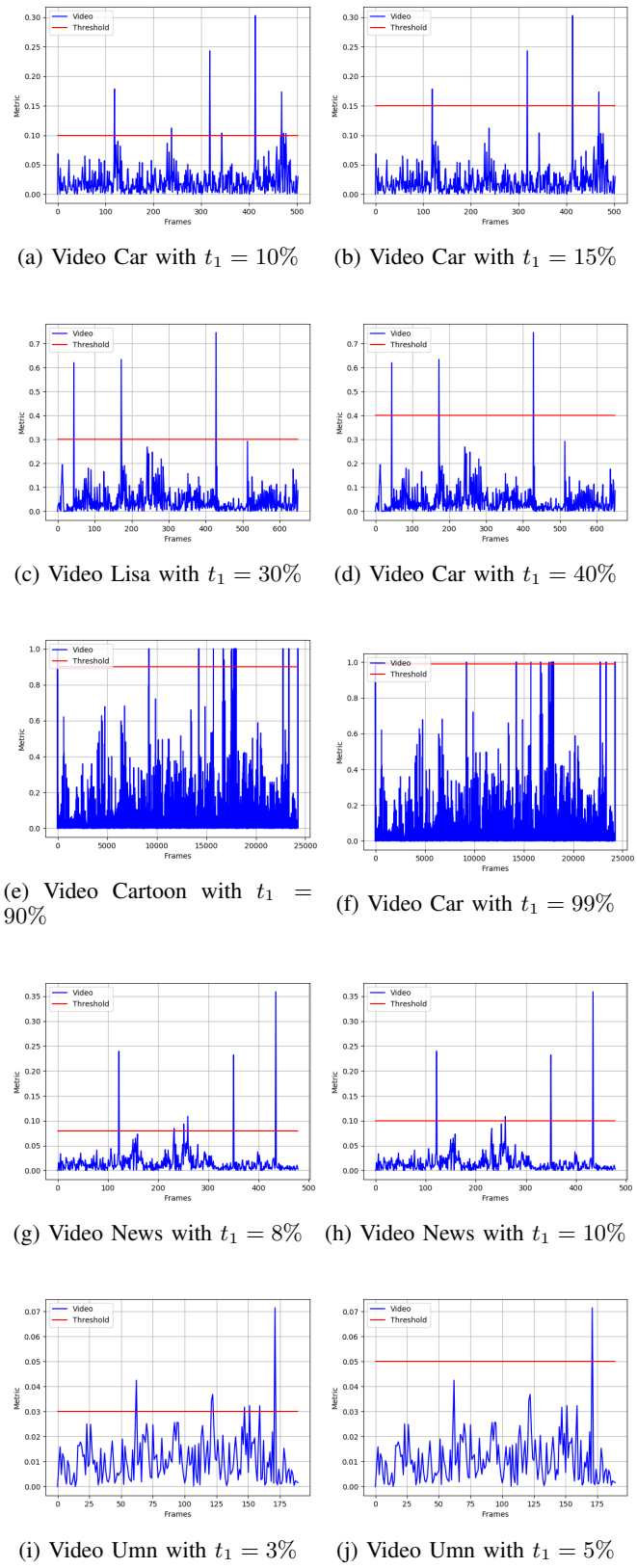
TABLE IV: Summary table of the experiments performed with the videos, show the thresholds and number of frames found that satisfy the restrictions.

the chosen metric, have value one, so for any chosen threshold, these will be cataloged as key frame. On the contrary the video Umn to be a video of a single scene, it is very difficult to make a summary, this because the abrupt changes are very few in this type of videos, but as shown in Figs. 8g and 8h, summarization is good with very small t_1 values.

In conclusion, this method is better that the others, because it stop using the intensities of the pixels to taken the decisions of when a frame is an abrupt transition, but in turn has a direct dependence on the shape of the image and the edge detectors used, which are also hyper parametrized algorithms, which includes more variables to the model.

III. Conclusion

In general all methods seen in this work, have good performance for problems where the environment variables are well controlled, the first three are methods where the calculation of the key frame is based on the values of the intensity of pixels, and the last method is based on the edges present in each frame.



(a) Video Car with $t_1 = 10\%$ (b) Video Car with $t_1 = 15\%$

(c) Video Lisa with $t_1 = 30\%$ (d) Video Car with $t_1 = 40\%$

(e) Video Cartoon with $t_1 = 90\%$ (f) Video Car with $t_1 = 99\%$

(g) Video News with $t_1 = 8\%$ (h) Video News with $t_1 = 10\%$

(i) Video Umn with $t_1 = 3\%$ (j) Video Umn with $t_1 = 5\%$

Fig. 9: Resulting graphs from the application of the edges difference process. The x axis shows the frames and the y axis show the metric difference used to categorize whether or not a frame is a change of shot.

All methods presented in this work are basic methods to solve the problem of video summarization, this can be seen when trying to perform the summarization of videos like Cartoon, which have a lot of changes throughout the video of both shots like tones within the same scene, so the more naive methods like these tend to respond poorly with this type of videos.

Checking all the experiments with the video Umn, we realized that with a good parameterization, we can get an acceptable summarization for videos where we always have the same background, where having small transitions with simple method like these we can get information of what is going on, this can be very helpful to surveillance videos.

In order to improve these methods, it is necessary after the extraction of key frames, to perform a small post processing, which allow us to find the optimal threshold for each big difference found in the video, which could be called an adaptive threshold. Also we propose the creation of another method, which uses the techniques of background subtraction mixed with the method of edge differences, we believe that this would allow to find with better certainty the key frames of the video.